REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
05/06/2005	Proceedings Paper/ Presentation (Unclassified)	05/06/2005-06/15/2005

4. TITLE AND SUBTITLE
Prognostics Models of Combat Vehicles

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Dr. Elena Bankowski and Dr. Abul Masrur

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
TARDEC, RDECOM
6501 E 11 Mile Road
Warren, MI 48397-5000

8. PERFORMING ORGANIZATION REPORT NUMBER
14926

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified/Unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The Next Generation Software and Survivability Technology areas of TARDEC RDECOM proposed the Dependable Automated Reconfigurable Technology (DART). The DART's "Health & Situation Control" will test the processing elements with Probe/Agent technology for software checking. Algorithms within the Health & Situation Control will assess the health of the processors and recommend element hand-off based on a "Criticality Scoring System" in conjunction with the Statistical Usage Test (SUT) model. The DART technology represents the next generation of software systems for ground combat vehicles. DART will enhance the performance of a weapon system by providing on-the-fly reconfiguration to accommodate the loss or malfunction of processing elements or to optimize onboard computational capability. Off-vehicle probes will be launched to assess the health of companion vehicles within the Operations Unit. The SUT will be used to evaluate software reliability. The SUT combined with a test environment that includes test benches, simulators and automated testing will provide the ability to arrive at a statistically valid

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Elena Bankowski
Unclassified	Unclassified	Unclassified	UU	7	19b. TELEPHONE NUMBER *(Include area code)* 586-574-6433

14926

SUPERIOR TECHNOLOGY FOR A SUPERIOR ARMY

TARDEC

IVSS-2005

Prognostics Models of Combat Vehicles Software

Dr. Elena Bankowski
Survivability Engineering Technology Area

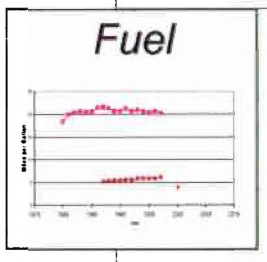
Dr. Abul Masrur
Next Generation Software Engineering Technology Area

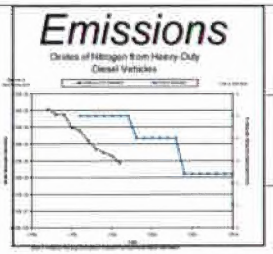
US Army RDECOM-TARDEC, Warren, MI 48397-5000

Software Intensive Feature Growth

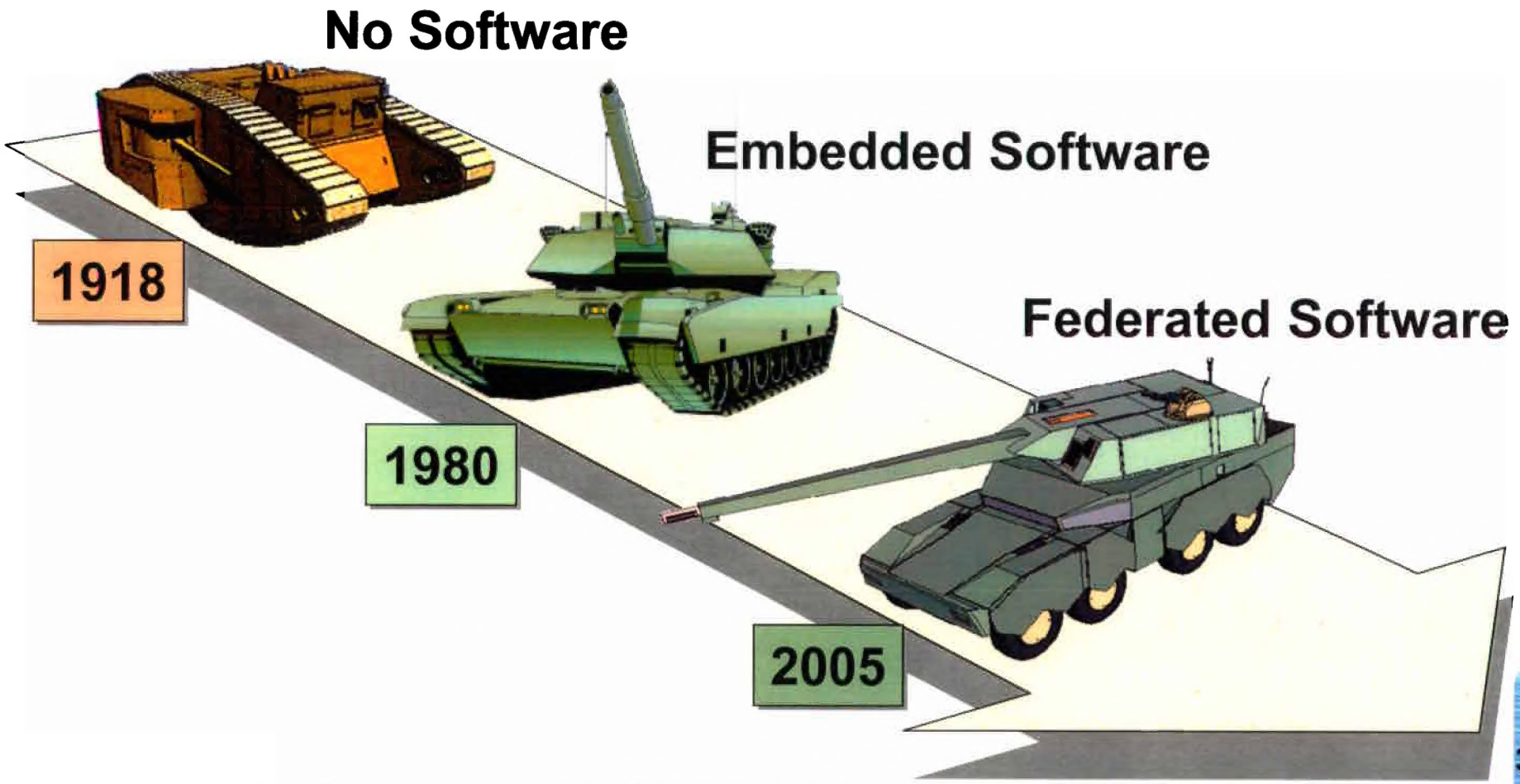
Improved Lean Logistics

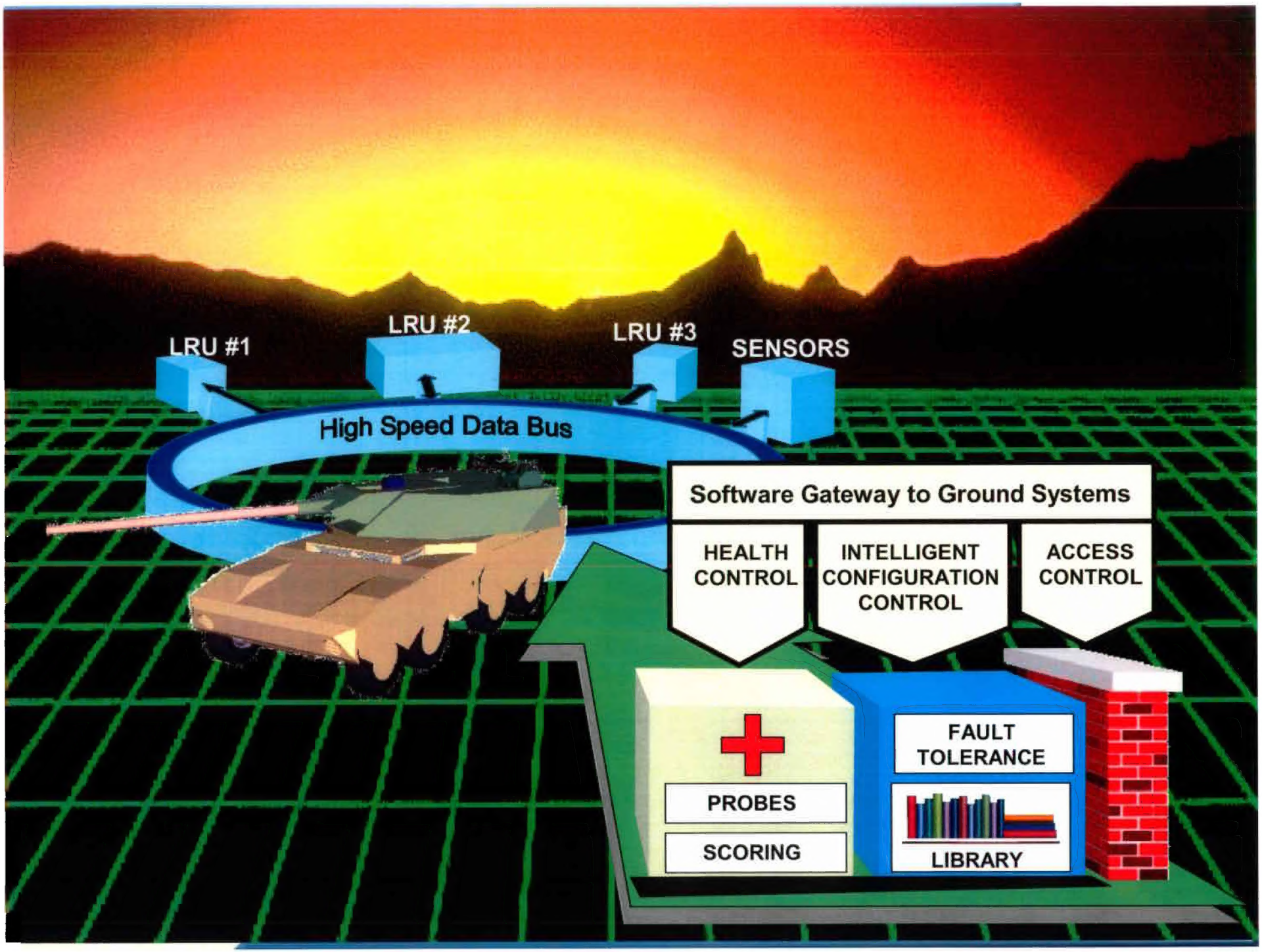
Reliability



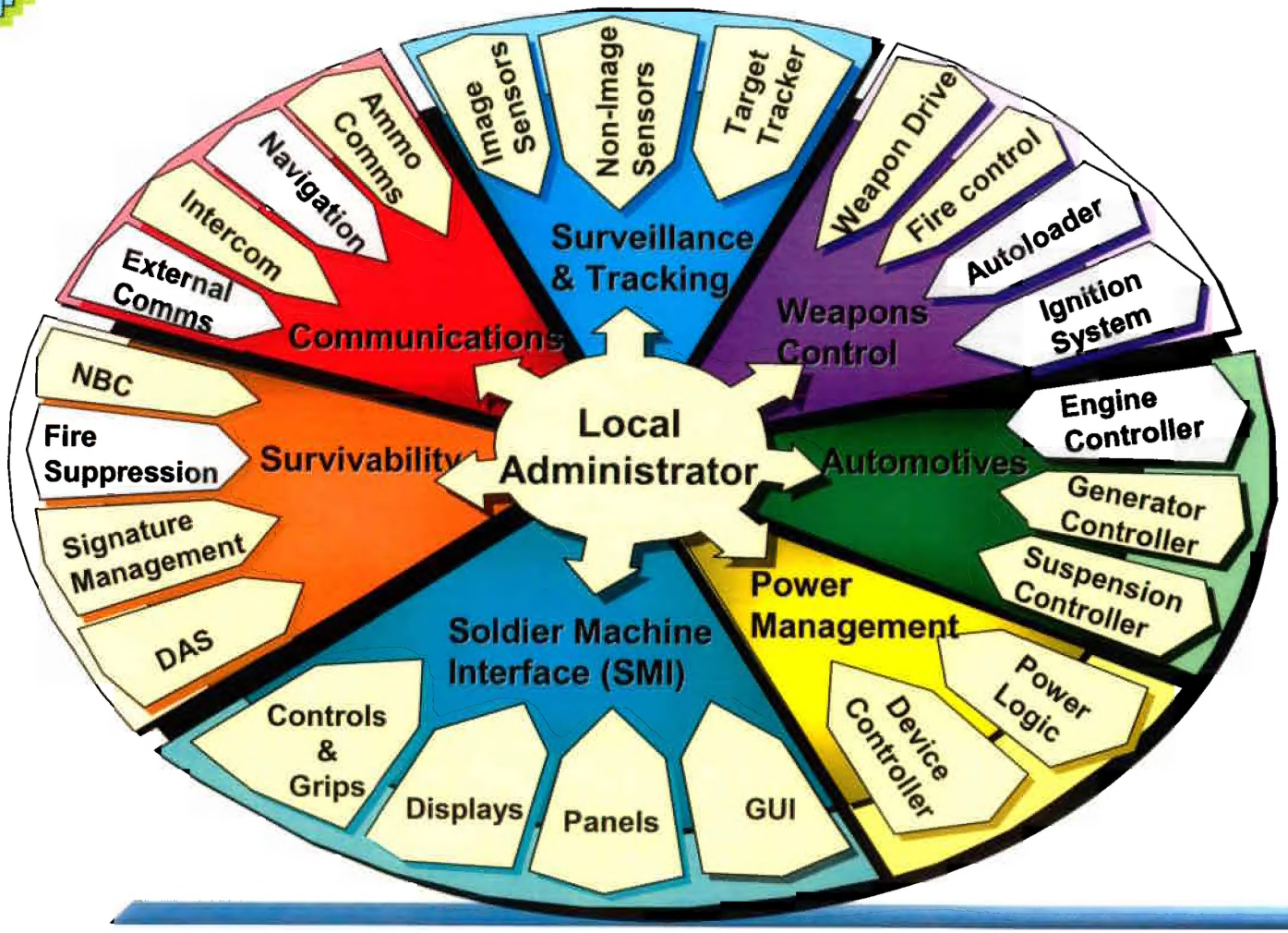


Software Evolution

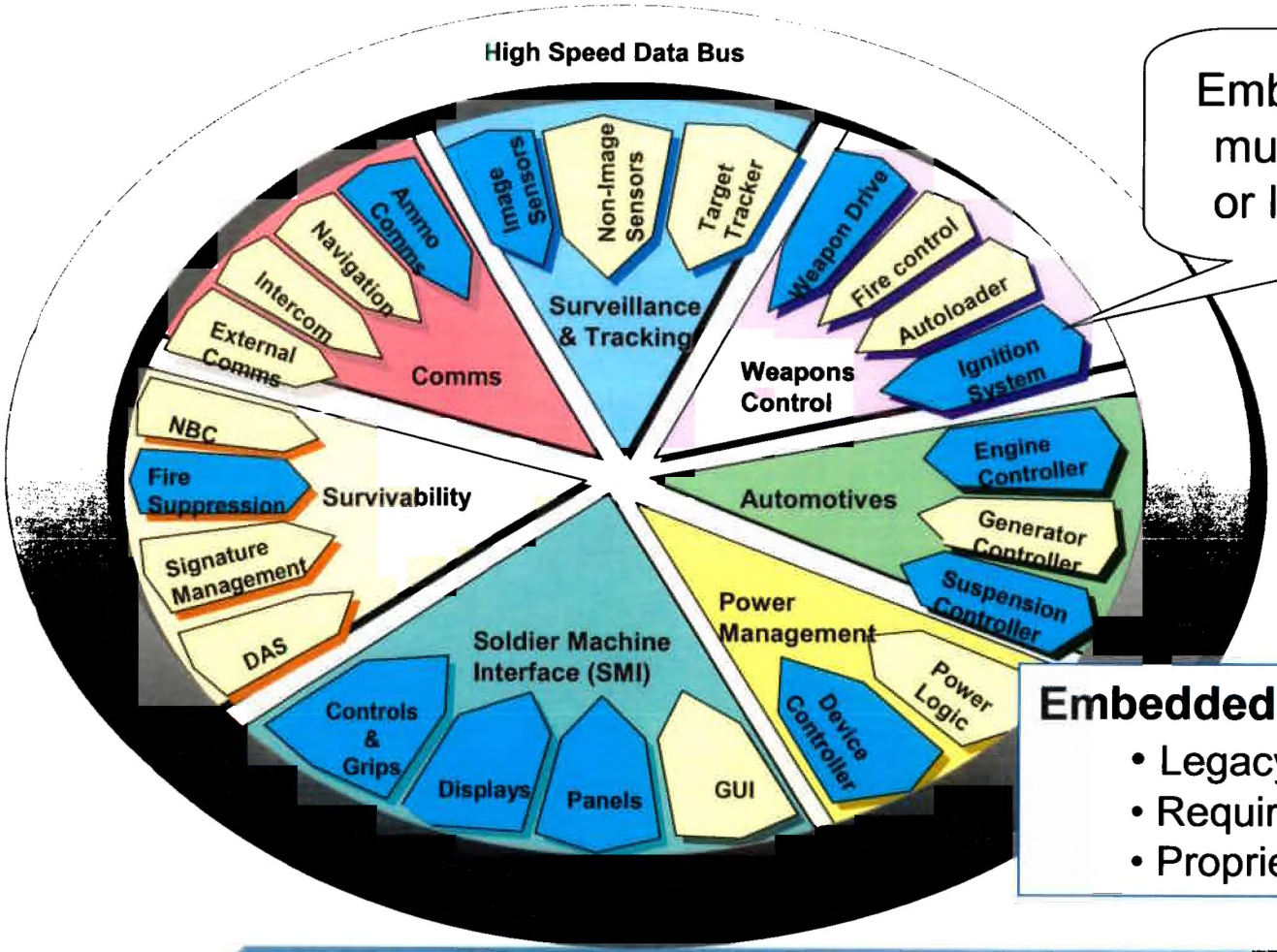



LRU #1
LRU #2
LRU #3
SENSORS
High Speed Data Bus
Software Gateway to Ground Systems
HEALTH CONTROL
INTELLIGENT CONFIGURATION CONTROL
ACCESS CONTROL
FAULT TOLERANCE
PROBES
SCORING
LIBRARY

Hub & Spoke Functional Categories of Software



Embedded Systems (shown in blue)

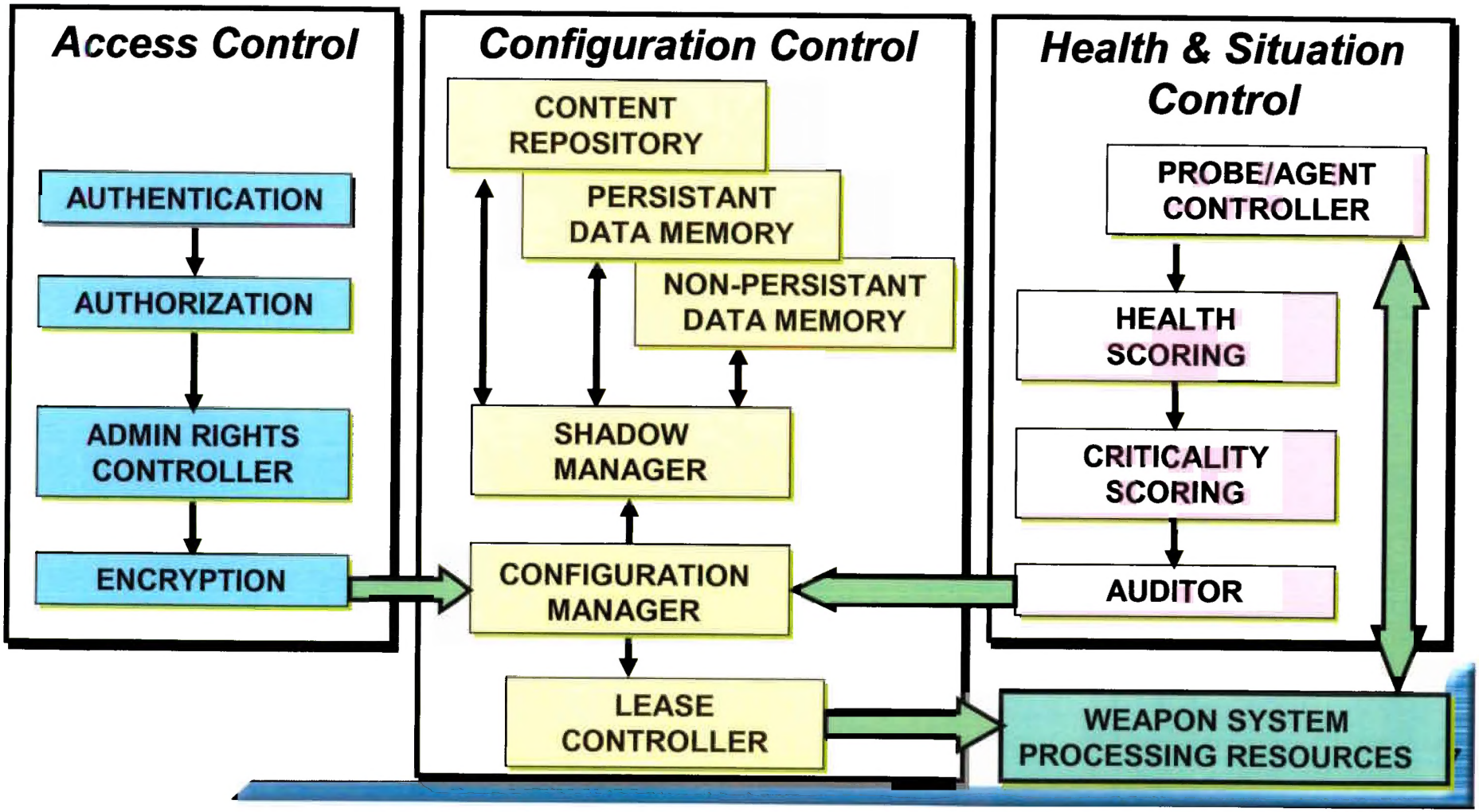


Embedded elements must be eliminated or limited in extent.

Embedded elements:

- Legacy Hardware/Software
- Required by Latency Issues
- Proprietary Sub-systems

Administrator's Operational Areas



TECHNICAL DISCUSSION

Probes

- Probes are able to capture events that occur in the software systems.
- These must span multiple software technologies used in the target software.
- They must be able to be inserted automatically to avoid software rework.
- A variety of probes is necessary in order to capture various kinds of information. The probes must have negligible impact on the behavior of the software.
- They must not interfere with operations by slowing response.
- The act of monitoring should not change the timing behavior, in testing it is misleading, in operational environments it could be catastrophic.
- The solution to this is three fold:

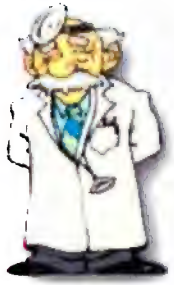
1. Probes must be engineered to have minimal space and performance impact.
2. Performance thresholds must be defined and monitored for critical software interactions.
3. It must be possible to turn off monitoring when these thresholds are in danger of being violated.

TECHNICAL DISCUSSION (Continued)

Gauges

Gauges analyze probed event streams and report summaries/conclusions in a more usable fashion. Gauges output can be used in four distinct ways:

- Gauges can feed back to previous software activities.
- Gauges can be used to affect previous software lifecycle steps.
- Gauges can control the operation of the target software, possible by triggering a reallocation of resources, shutting down nonessential functionality, and even compensating for errors.
- Gauges can control the data collection activities by activating and deactivating probes to avoid impacting performance or to focus collection activities on suspect or critical areas.



Health and Situation Control

PROBE/AGENT CONTROLLER

This activity will determine the frequency, format and type of query that the local administrator will send to the elements so that element health can be determined.

HEALTH SCORING

This activity will evaluate processed data to determine if it lies within the output bands of the element

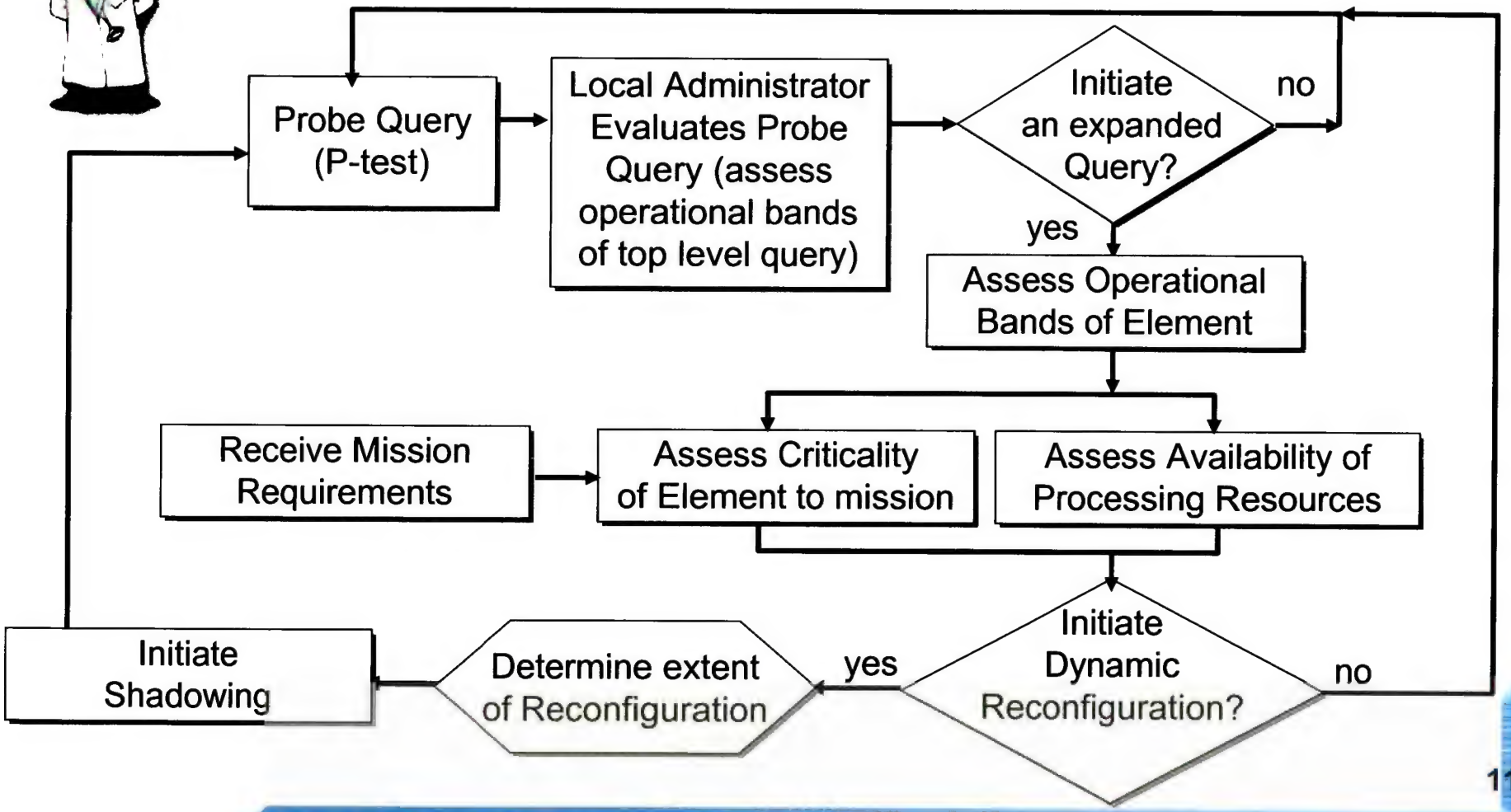
CRITICALITY SCORING

Based on the nature of the mission, and the battlefield situation, this activity will prioritize vehicle functions that require processing

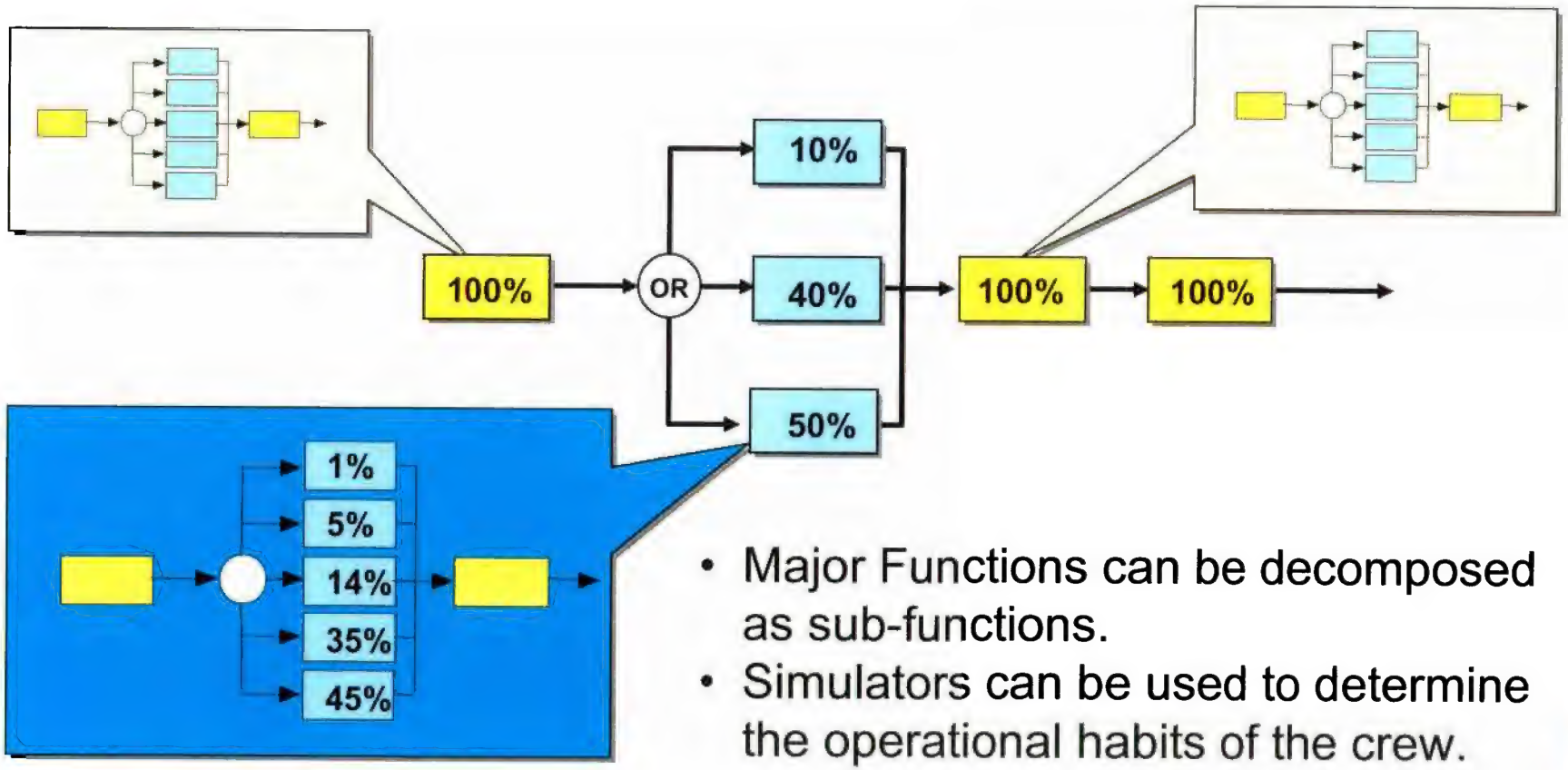
AUDITOR

This activity will compile and report the collective operational and health status of all of the elements to the local administrator

System Health Check Process

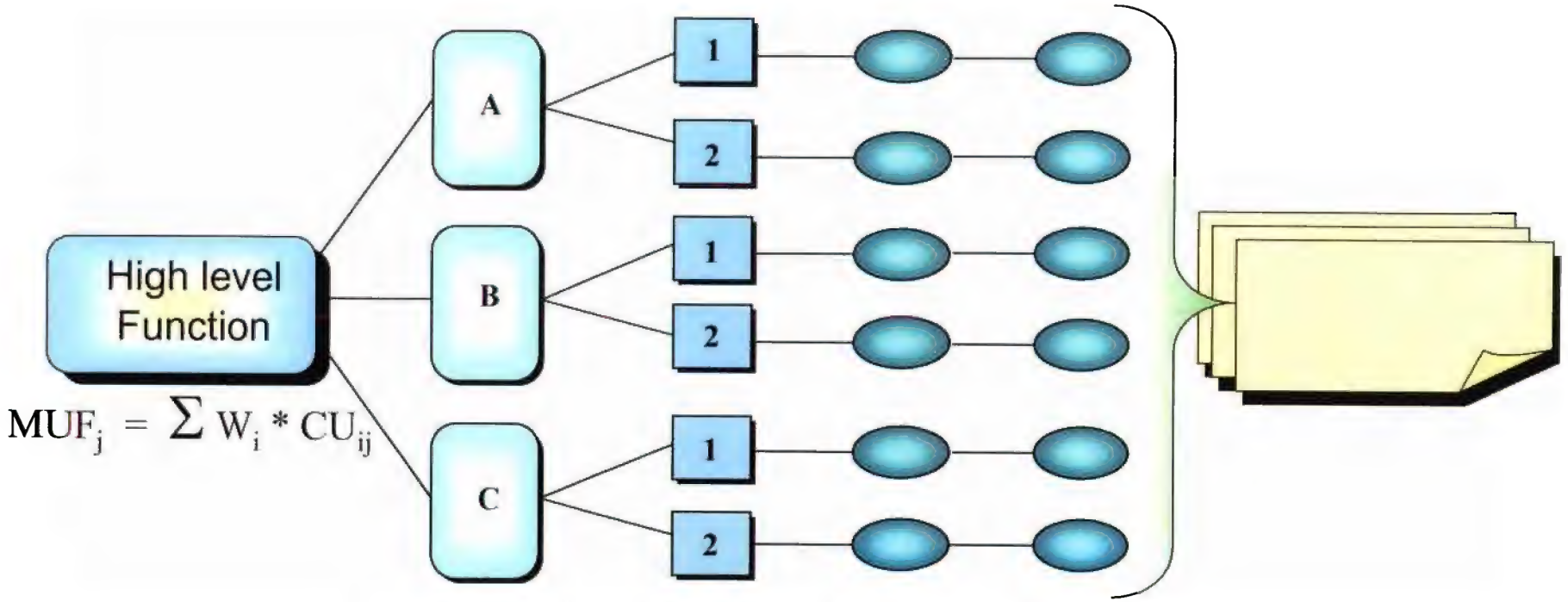


Functional Flow Block Diagram (for evaluation of usage)



- Major Functions can be decomposed as sub-functions.
- Simulators can be used to determine the operational habits of the crew.

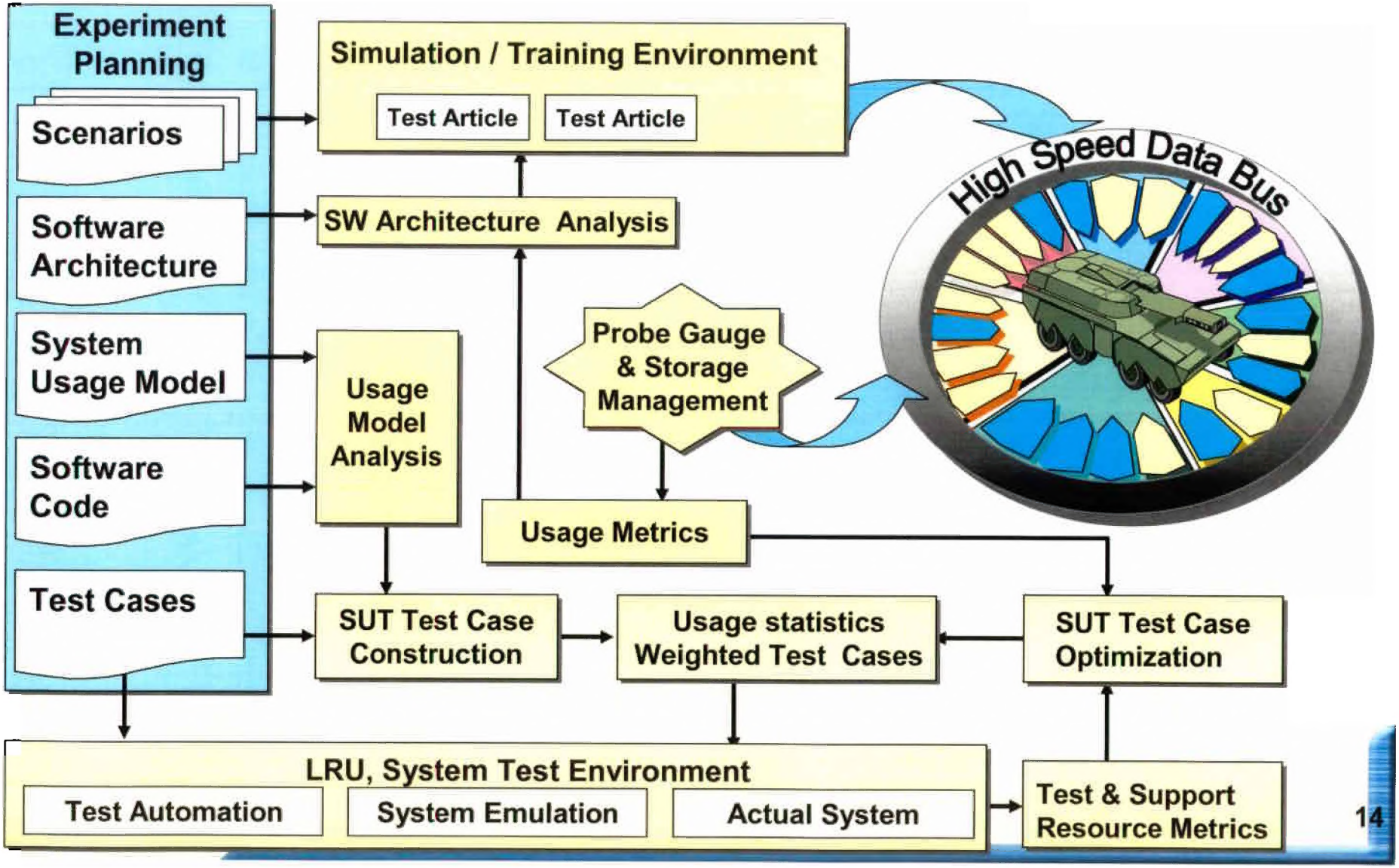
Criticality Evaluation of Element



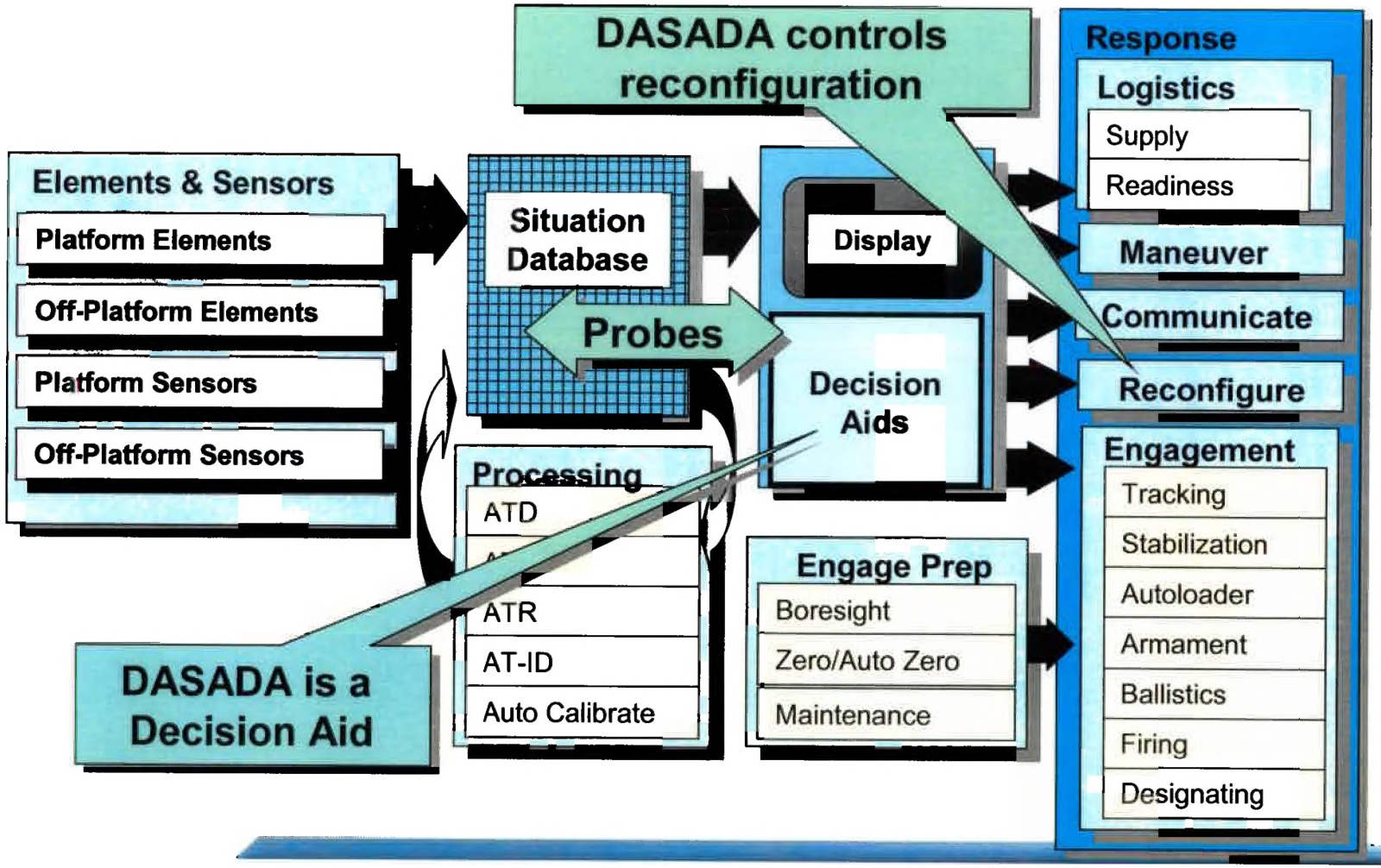
$$MUF_j = \sum W_i * CU_{ij}$$

W_i

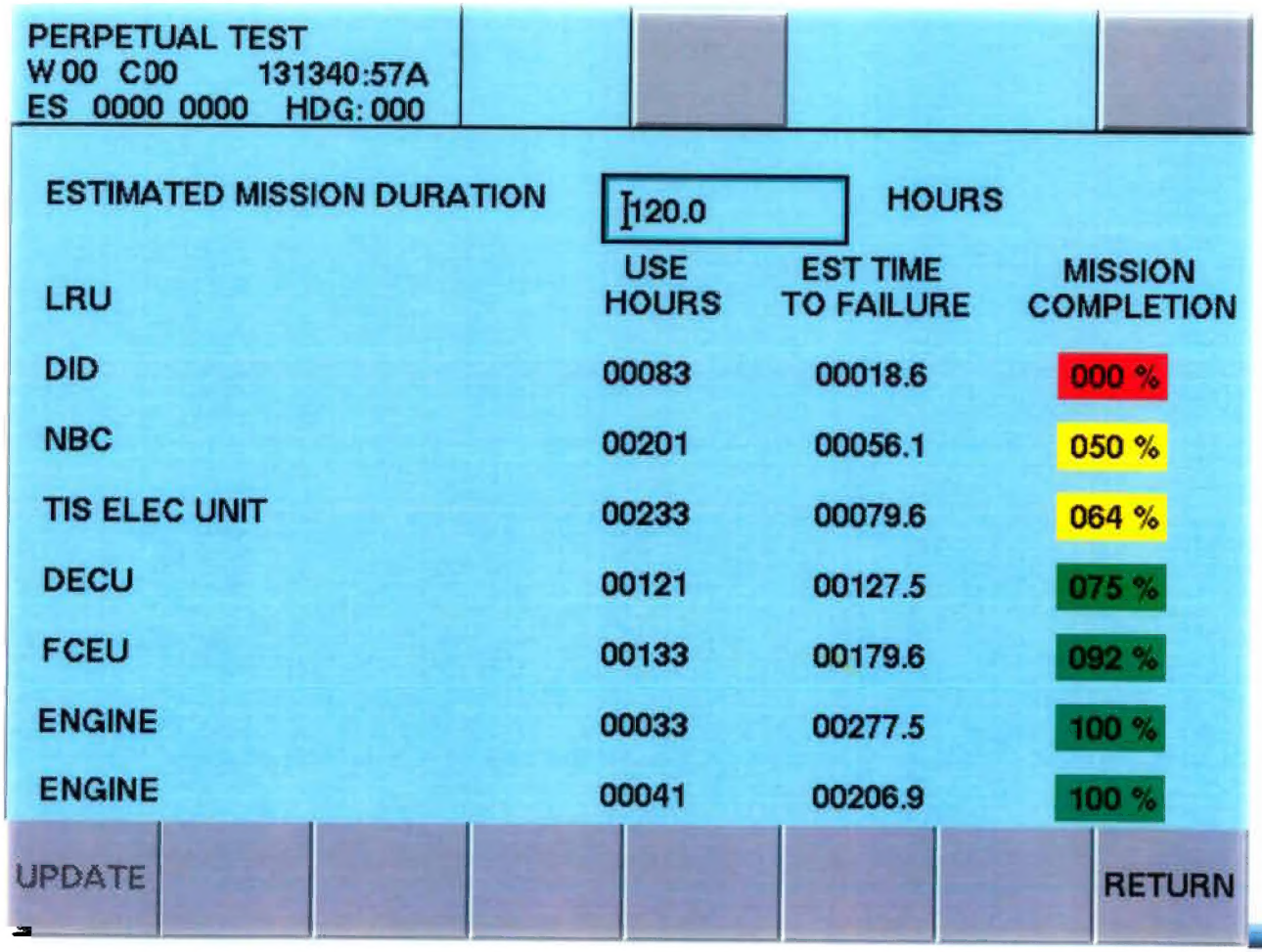
$CU_{ij} = SUF_i(ML_{ij})$


Experiment Planning
Scenarios
Software Architecture
System Usage Model
Software Code
Test Cases
Simulation / Training Environment
Test Article
Test Article
SW Architecture Analysis
Usage Model Analysis
Probe Gauge & Storage Management
High Speed Data Bus
Usage Metrics
SUT Test Case Construction
Usage statistics Weighted Test Cases
SUT Test Case Optimization
LRU, System Test Environment
Test Automation
System Emulation
Actual System
Test & Support Resource Metrics

DASADA and the FCS Architecture



Proposed Prognostic Screen



The DART Process